**Practical 7**

**Assembly Language**

NOTE: Use of EASy68K editor and emulator allowed, use of internet allowed, use of slide deck(s) allowed. Installer located here [**http://tinyurl.com/Easy68K**](http://tinyurl.com/Easy68K)  Create a unique file ***e.g. part1.X68*** for each practical section below.

**Objective** Understand and utilise Assembly **Addressing Modes:**

(1) Direct (2) Immediate (3) Indirect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Create a new 68K project and name the file ***part1.X68***    Edit compile and execute the code across, examine and note contents of data registers and memory. Identify the memory location of 100 and its contents. | | MOVE.B  MOVE.B  MOVE.B  MOVE.B  MOVE.B | #9,D1  D1,D2  D1,1100  1100,D2  1100,100 | |
| 2. Create a new 68K project and name the file ***part2.X68***    Edit compile and execute the code across and observe the output. | | ORG $1000  START:  MOVE.B #9,D1  LEA text, A1  MOVE #14,D0  TRAP #15    MOVE #3,D0  TRAP #15    SIMHALT    text dc.b 'Data Register: ',0    END START | | |
| 3. Create a new 68K project and name the file ***part3.X68***    Edit compile and execute the code across and observe the input and output. | ORG $1000  START:  LEA text, A1  MOVE #4,D0  TRAP #15  MOVE #14,D0  TRAP #15  MOVE #3,D0  TRAP #15 | | |

SIMHALT

text dc.b 'Data Register: ',0

# END START

**Q1:**

**Calendar

Description automatically generated**

This set of instructions moved (via move-by-copy) copied the value 9 from many data registers and locations. The last instruction copied the value 9 from the memory location 44C (which is 1100 in decimal) to the memory location 64 (which is 100 in decimal).

**Q2:**

**Graphical user interface, text, application

Description automatically generated**

Final output

**Text

Description automatically generated**The Trap #15 read instruction (or “Task”) #14 in Data Register 0 to display the value at the address stored in Address Register 1. This would mean that Task 14 implicitly dereferences the address. Below, we can see the address of the text variable being 101A and that the hex representation of the text string ranges from 1010 to 1019

**A picture containing diagram

Description automatically generated**

**Q3**

**Graphical user interface, text, application, email

Description automatically generated**

This program defines the variable text which stores the string “Data Register: “, and places it in the Address Register 1. It then stores the value 4 to be treated as Task 4 (take input from the user’s keyboard and store it in Data Register 1) in the upcoming Trap Instruction. After which it then moves the value 14 to be treated as Task 14 (display the string value at the address stored in Address Register 1) for the second Trap instruction while the next Task to be executed is Task 3 (display the signed number from Data Register 1 in the smallest possible width).

Below we entered the negative sign -123 where you can see all the “unused” bits flipped to F to indicate -123 whereas in the positive value in the second image, all of the unused bits stay at 0 to indicate positive.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

**Demonstrate completed assembly files at the end of the LAB and ensure it has been checked**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **David Darigan** | **Student Number** | **C00263218** |
| **Date** | **12/01/2022** | **Checked** |  |

Page **1** of **1**